30th NATIONAL RADIO SCIENCE CONFERENCE
(NRSC 2013)
April 16-18, 2013, National Telecommunication Institute, Egypt

C24. An Enhanced Particle / Kalman Filter for Robot Localization

Imbaby I. Mahmoud [1]*, Asmaa Abd El Tawab*[2] *and , May Salama*

[1]Nuclear Research Center, Atomic Energy Authority,Imbabyisma@yahoo.com
[2]Nuclear Research Center, Atomic Energy Authority,asm28aa@yahoo.com
[3]Benha University, Faculty of Eng., Shoubra , Eletronic and Comm. Dept.,msalama@megacom-int.*com*

ABSTRACT

This paper investigates the effect of using different filters namely: Kalman filter (KF), Particle Filter (PF) and a proposed enhanced particle / Kalman (EPKF) filter based robot localizer. An algorithm is built in Matlab environment to host these filters. The performances of these filters are evaluated in terms of computational time and error from ground truth and the results are reported. The results showed that the proposed localization plan which adopts the particle filter as initialization step to Kalman filter achieves higher accuracy localization while, the computational cost is not significant.

Keywords: *Particle Filter, Kalman filter, Robot Localization*

I. INTRODUCTION

The problem of robot localization consists of answering the question Where am I from a robot's point of view. This means the robot has to find out its location relative to the environment. When we talk about location, pose, or position we mean the x and y coordinates and heading direction of a robot in a global coordinate system.

The mobile robot localization problem comes in many different flavours .The most simple localization problem is position tracking while the initial robot pose is known, and the problem is to compensate incremental errors in a robot’s odometry. More challenging is the global localization problem [1], where a robot is not told its initial pose but instead has to determine it from scratch.

Several methods are employed to deal with robot localization problem [2, 3] .The Kalman filter has been frequently applied to the problem of robot localization. It works recursively, and so does not require a history of the robot’s previous states to be kept. This results in a streamlined algorithm that can run online in a real time systems. Unfortunately, the absolute range measurements are non-linear (as in our case) requiring the use of an extended Kalman Filter (EKF), which must be linearized the measurements around the current state estimate. These results in a weakness common to all linear methods which means that the Kalman filter will not converge when the initial state is not sufficiently accurate [4]

Recently Particle Filter (PF) becomes dominant approach used for solving this problem. This is due to its ability to handle non-linear non-Gaussian problem, typical characteristic of localization problem [5, 6]. Several implementations of PF are reported [7- 9].

In this paper, the particle filter is introduced to initialize Kalman filter to overcome the initial state problem of original Kalman filter. Different filters namely Kalman filter (KF) , Particle Filter (PF) and a proposed enhanced particle/Kalman (PKF) implemented in Matlab environment and their performance are evaluated in terms of computational complexity and amount of error from ground truth The obtained results are reported and compared..

This paper is organized as follows: section2 presents overview of an Enhanced Particle / Kalman filter and their implementation algorithms, section 3 studied the effect in Robot Localization by using different filters, section 4 shows the discussion of the obtained results, section 5 is devoted to conclusion.

II. OVERVIEW OF AN ENHANCED PARTICLE / KALMAN FILTER

The used absolute range measurements are non-linear, requiring the use of an *extended* Kalman Filter (EKF), The Kalman filter there is modified to filter known as extended Kalman filter

978-1-4673-6222-1/13/$31.00 ©2013 IEEE

A. Extended Kalman Filter (EKF) for Localization

1) Process Model [10]

If the robots pose (position and attitude) at time *k* is represented by the state vector $q_k = [x_k, y_k, \theta_k]^T$ then the dynamics of the wheeled robot used in this experiment are well-modeled by the following set of non-linear equations:

$$q_{k+1} = \begin{bmatrix} X_k + \Delta D_k \cos(\theta_k) \\ y_k + \Delta D_k \sin(\theta_k) \\ \theta_k + \Delta\theta_k \end{bmatrix} + v_k \tag{1}$$

Where: v_k is a noise vector. Here, ΔD_k point at the center of the robot's front axle, obtained by averaging the distances measured by the left and right wheel encoders. The incremental orientation change *Δθk* is obtained by the onboard gyro. These dead reckoning measurements constitute the control input vector $u_k = [\Delta D_k, \Delta\theta_k]^T$
The system matrix *A (k)* is represented by the Jacobian:

$$A(k+1) = \frac{\partial f}{\partial q_k}|_{q-\hat{q}_k} = \begin{bmatrix} 1 & 0 & -\Delta D_k \sin(\theta_k) \\ 0 & 1 & \Delta D_k \cos(\theta_k) \\ 1 & 0 & 1 \end{bmatrix} \tag{2}$$

The input gain matrix *B(k)* is constructed similarly:

$$B(k+1) = \frac{\partial f}{\partial q_k}|_{q-\hat{q}_k} = \begin{bmatrix} \cos(\theta_k) & 0 \\ \sin(\theta_k) & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

2) *Measurement Model [10]:*

The range at time *k+1* from a beacon located at *(xb, yb)* to the robot with state vector q_{k+1} can be expressed as:

$$h\left(q_{k+1,} [x_{b,} y_b]^T\right) = \sqrt{(x_{k+1} - x_b)^2 + (y_{k+1} - y_b)^2} \tag{4}$$

▤➂ *Time Propagation [11]*

When a new control input vector $u(k) = [\Delta D_{k,} \Delta\theta_k]$ is received, the robot's state is updated according to the process model equation. Using the standard equations of Kalman filtering, the covariance matrix maintaining our uncertainty about the current state is propagated in time:

$$p_k^- = A(k) p_{k-1}^+ A(k)^T + B(k)\Gamma B(k)^T + Q(k) \tag{5}$$

So, the state maintained during the time propagation step indicates the pose of the robot at the robot reference point.

4) *Measurement Update: [12]*

When a measurement is obtained, `using the method of` the update step is broken up as follows:

1. Using the current state estimate, determine the location of the antenna which received the current measurement (i.e., shift the robot reference point's coordinates to get the coordinates of the current antenna, *(xa, ya)*).
2. Project the current measurement onto the *xy* plane of the robot.
3. Using *(xa, ya)* and the known beacon location *(xb, yb)*, compute *Hk*.
4. Look up the variance *k R* and the mean *k y* associated with the current measurement from its stored PDF.
. Using the measurement model, compute the expected range *rk* to the beacon. Let υ (*k*) = *y* − *r* be the innovation.

6. Compute $S_k = H_k p_k^- H_k^T + R_k$
7. Compute the Kalman gain $K_k = p_k^- H_k^T S_k^{-1}{}_k$
8. Compute the normalized innovation squared and test the measurement against the chi square
9. If the measurement passes the gating test, update the state by letting $\hat{q}_k^+ = \hat{q}_k^- + K_k v(k)$ and update the covariance matrix by letting $p_k^+ = p_k^- - K_k S_k K_k^T$.
10. Now, using this updated estimate of the pose at the antenna which reported the current measurement, shift back in *x* and *y* to get the updated pose estimate at the robot reference point.

B. Particle Filter Algorithm

The PFs are formulated on the concepts of the Bayesian theory and the sequential importance-sampling which are very effective in dealing with non-Gaussian and non-linear problems [13-14]

The PF approximates recursively the posterior distribution using a finite set of weighted samples. The idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. PF uses the probabilistic system transition model $p(X_t|X_{t-1})$, (which describes the transition for state vector X_t) to predict the posterior at time *t* as:

$$p(X_t \mid Z_{1:t-1}) = \int p(X_t \mid X_{t-1})\, p(X_{t-1} \mid Z_{1:t-1})\, dX_{t-1} \tag{6}$$

Where $Z_{1:t-1} = \{Z_1, Z_2, \ldots Z_{t-1}\}$ are available observations at times 1, 2, …., t-1, $p(X_t|X_{t-1})$ expresses the motion model, $p(X_{t-1}|Z_{1:t-1})$ is posterior probability density function at time t-1 and $p(X_t|Z_{1:t-1})$ is the prior Probability Density Function (PDF) at time t. At time *t*, the observation Z_t is available, then the state can be updated using Bayes's rule as:

$$p(X_t \mid Z_{1:t}) = \frac{p(Z_t \mid X_t)\, p(X_t \mid Z_{1:t-1})}{p(Z_t \mid Z_{1:t-1})} \tag{7}$$

Where $p(Z_t|X_t)$ is described by the observation equation. The posterior PDF $p(X_{t-1}|Z_{t-1})$ is approximated recursively as a set of *N* weighted samples $\left\{X_{t-1}^{(s)}, W_{t-1}^{(s)}\right\}_{s=1}^{N}$, and $W_{t-1}^{(s)}$ is the weight for particle $X_{t-1}^{(s)}$. Using a Monte Carlo approximation of the integral, we get:

$$p(X_t|Z_t) = p(Z_t|X_t) \sum_{s=1}^{N} W_{t-1}^{(s)}\, p(X_t \mid X_{t-1}^{(s)}) \tag{8}$$

The *N* samples $X_t^{(s)}$ are drawn from the proposal distribution:

$$q(X) = \sum_{s=1}^{N} W_{t-1}^{(s)}\, p(X_t| X_{t-1}^{(s)}) \tag{9}$$

Then it is weighted by the likelihood.

$$W_t^{(s)} = p(Z_t| X_t^{(s)}) \tag{10}$$

This produces a weighted particle approximation $\left\{X_{t-1}^{(s)}, W_{t-1}^{(s)}\right\}_{s=1}^{N}$ for the posterior PDF $p(X_t|Z_t)$ at time *t*.

1) Problem Formulation

This particle filter estimates the robot's planar position only, not orientation. So, each particle is a point in the state space (in this case the *xy* plane) and represents a particular solution. The primary reference for this filter is the paper on Condensation by Blake [15], 1998.In the following basic steps forming the particle filter for localization problem are investigated.

2) Initialization:

We choose the number of particles to be N=1000, with each particle initialized to a random state $\begin{bmatrix} X \\ y \end{bmatrix}$, so the particles are initially evenly spread over the *xy* plan within set boundaries. The probability of the *i*th particle is $\left[\frac{1}{N}\right]$, and a cumulative probability distribution is maintained for the particles as well. The cumulative probability for the *i*th particle is $\left[\frac{i}{N}\right]$.

3) Drift:

The drift is simply computed as the translation in *x* and *y* from the integrated dead reckoning path. The dead reckoning measurements are treated as a simple (*x, y*) translation since the particles are not oriented. So, at each time step the particles all drift by the same amount.

4) Diffusion:

After drifting, a random number is added to each particle's coordinates. A diffusion rate of B=0.03 m/s is chosen, so the random diffusion amount is scaled by B*(Δt), where Δt is the amount of time since the last measurement.

5) Sampling:

A probability is assigned to each of the particles according to the standard Gaussian formula:

$$p(q \mid r_m) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{\hat{r}-r_m}{\sigma}\right)^2} \tag{11}$$

Where *rm* is the range measurement
*r*ˆ is the range estimate from the particle to the beacon,
and σ is the standard deviation from the PDF for that range measurement.

6) Resampling :

The particles are resampled to represent their current probability distribution. This is done by the standard methods as resampling approaches namely : Multinomial Resampling (Mult R), Systematic Resampling (SR), Residual Resampling (RR), Residual Systematic Resampling (RSR) and Stratified Resampling (STR) the particles which has a large weight generate a number of copies which are going to be propagated to the next generation proportional to its weight[16].

III. Implementation and Results of Robot Localization Algorithm using Studied Filters

We studied a localization system which employs radio beacons that provide the ability to measure range only [17]. Obtaining range from radio beacons has the advantage that line of sight between the beacons and the transponder is not required, and the data association problem can be completely avoided. In this work seven radio beacons are distributed over two different areas robot is programmed to drive in a repeating path. All filters methods are used to fuse range data with dead reckoning data collected from a real system which integrates proprioceptive measurements from wheel encoders, gyros, and accelerometers to localize the robot. Matlab environment is used for experimenting with localization process.

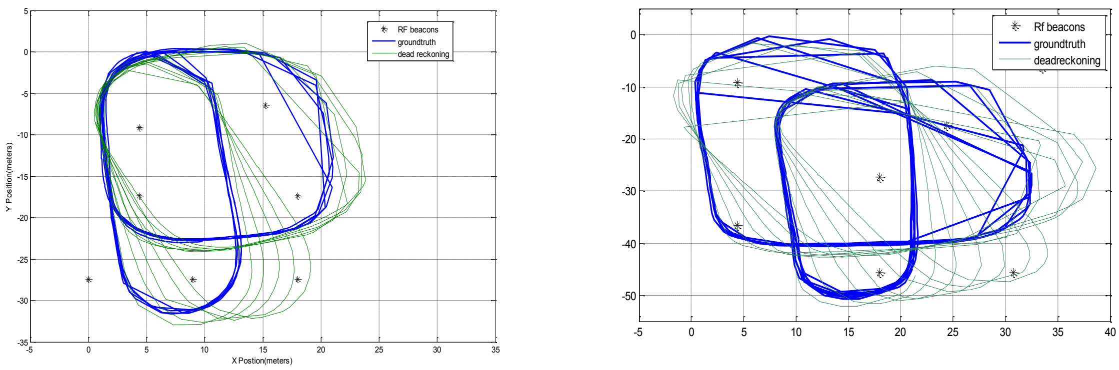


Fig .1: The ground truth path, tag locations and dead reckoning from left to right. (a) The first dataset (A1), (b) the second dataset (A2)

Fig [1] shows the dead reckoning path, ground truth path and tag locations for the first path dataset [A1] in. We notice from these figures that the dead reckoning tends to drift away from the true path over time. This is due to increasing errors in odometry.

A. The results of the different approaches

In this paper we have presented two carefully-collected datasets and processed them with an extended Kalman filter, a particle filter, and Enhanced particle / Kalman filter. Our implementation of particle in matlab environment requires no initial estimate of the robot's position. In all experiments, the robot's travel is clipped from results plot, giving the filter time to converge. Figs (2-4) show the results of studied filters. Table [1] & [2] summarizes the results of these figures concerning the error in the estimates of the studied filters.

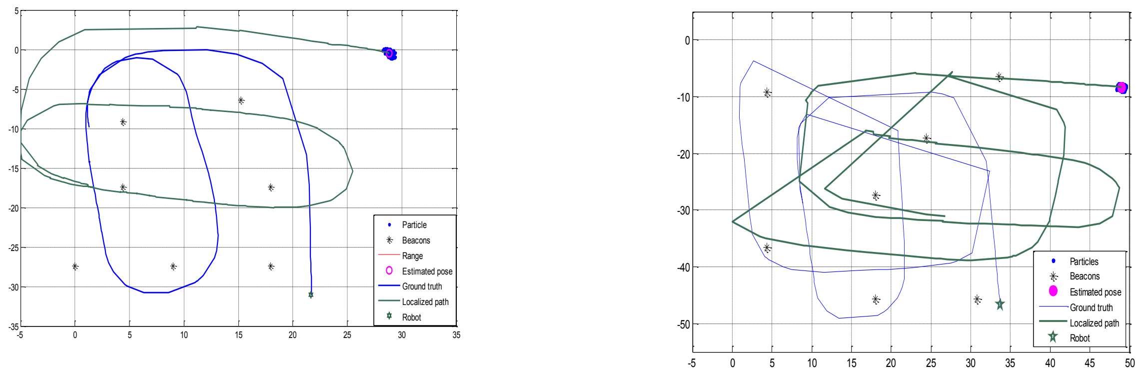


Fig.2: particle filters localization performance on a) first dataset (A1) b) Second dataset (A2).

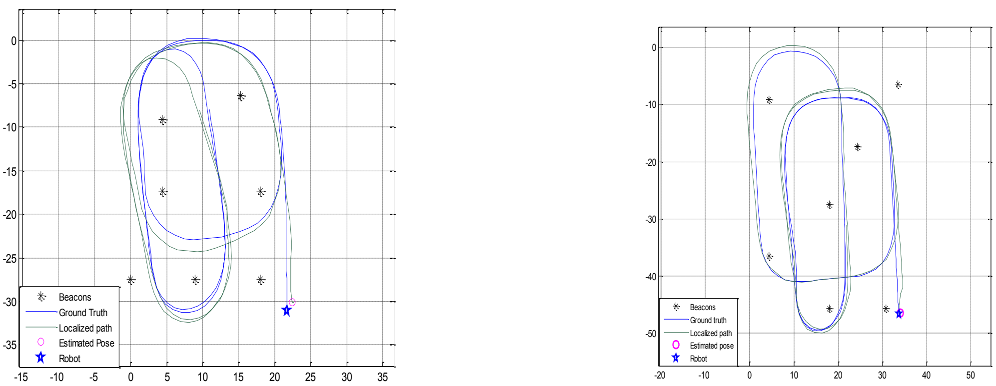
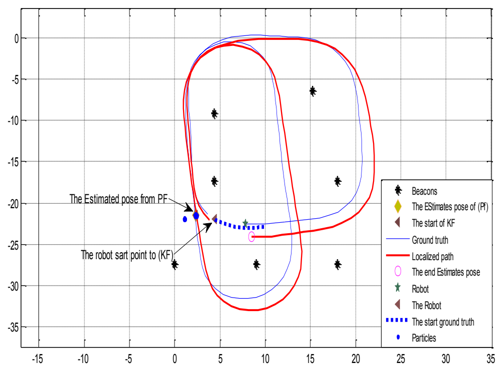


Fig.3: Kalman filter localization performance on a) first dataset (A1) b) Second dataset (A2).



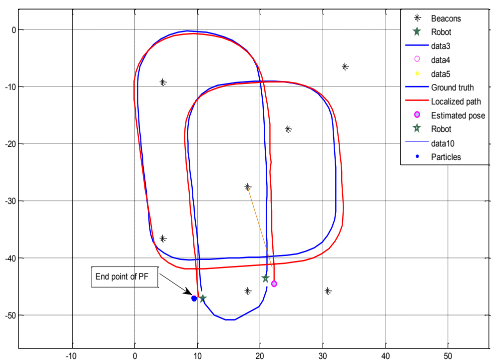


Fig. 4: Enhanced EPKF localization performance on a) first dataset (A1) b) Second dataset (A2) at the connecting steps between the two filters.

Table 1: Results of Error Calculation Using Different Filters in the first Dataset (A1)

Error/in meter	PF	EKF	PKF
XTE_abs_avg	3.5883	0.8787	0.8841
XTE_abs_max	23.4162	2.5697	2.5673
XTE_abs_std	3.622	0.5946	0.5983
ATE_abs_avg	5.3397	1.1241	1.1386
ATE_abs_max	32.7119	3.5205	3.5204
ATE_abs_std	5.0402	0.792	0.7908
Cartesian_abs_avg	7.0857	1.5502	1.5697
Cartesian_abs_max	33.3154	3.5315	3.5314
Cartesian_abs_std	5.4501	0.7748	0.7662

XTE:Cross Track Error, How far left or right of the true position our estimation is, Orthogonal to the true heading, ATE: Along Track Error, Tangential component of the position error ,Cartesian error: Total Euclidean distance error /in meter .

Table 2: Results of Error Calculation Using Different Filters in the first Path Data (A2)

Error/in meter	PF	EKF	PKF
XTE_abs_avg	7.2004	0.6052	0.6119
XTE_abs_max	36.9777	1.8401	1.7059
XTE_abs_std	7.3529	0.3987	0.3952
ATE_abs_avg	8.768	0.5405	0.5368
ATE_abs_max	37.3803	1.6589	1.7392
ATE_abs_std	8.3345	0.3644	0.3603
Cartesian_abs_avg	12.5861	0.8862	0.8882
Cartesian_abs_max	40.9376	1.8673	1.8283
Cartesian_abs_std	9.6871	0.402	0.3996

IV. Discussion of Results

The results are summarized graphically using bar chart in Fig [5-6]. Chiefly, we consider the cross-track error (abbreviated XTE), which gives the component of position error that is orthogonal to the robot's path. We also present the along-track error (abbreviated ATE), which measures the tangential component of position error.

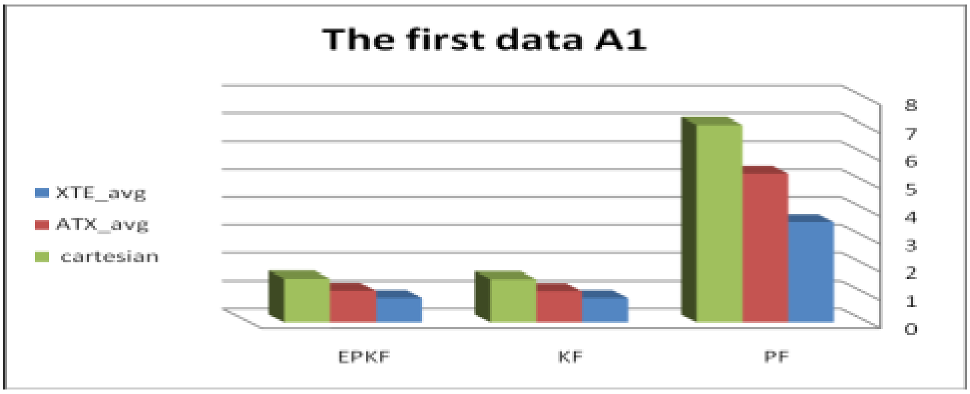


Fig. 5: The EKF, the Enhanced EPKF and PF along-track error, the cross-track error and Cartesian error

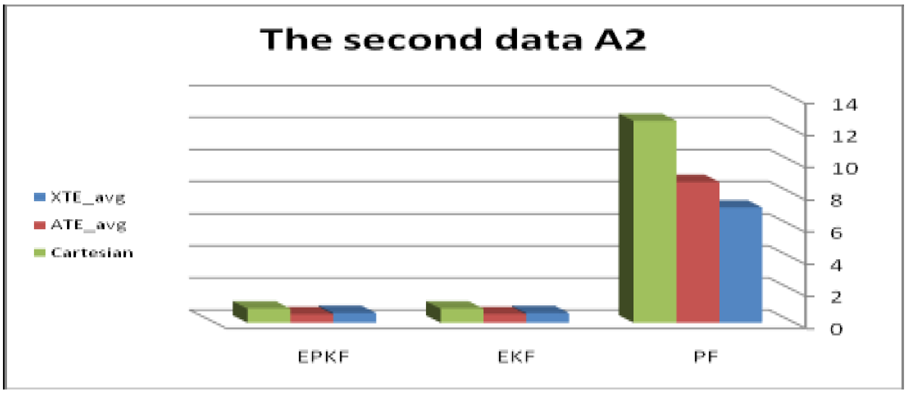


Fig.6: The EKF, the Enhanced EPKF and PF along-track error, the cross-track error and Cartesian error

From Table [1] & [2] and fig 5-6 comparable results we notice the slight difference in calculated error among extended Kalman filter and the proposed Enhanced particle / Kalman filter while the particle filter posses excessive error.

Considering computational complexity and time consumed in a Matlab run, Fig [7] shows the time consumed by each filter in the same environmental Conditions. There is a slight increase in time for the propose EPKF compared with EKF while the PF consumes higher time. Therefore, the proposed filter achieves the same results of EKF while keeping the computational cost reasonable and via the time solving the problem inherent of all Kalman filters which require a defined initial state.

Table.3 Summarizes the Average Time each Algorithm Requires to Incorporate an Incoming Range Measurement into the Robot Position Estimate

Running Times	seconds per measurement update
Particle Filter	0.142494
EKF	0.007988
EPKF: PF estimate an initial state which to seed the EKF.	0.014385

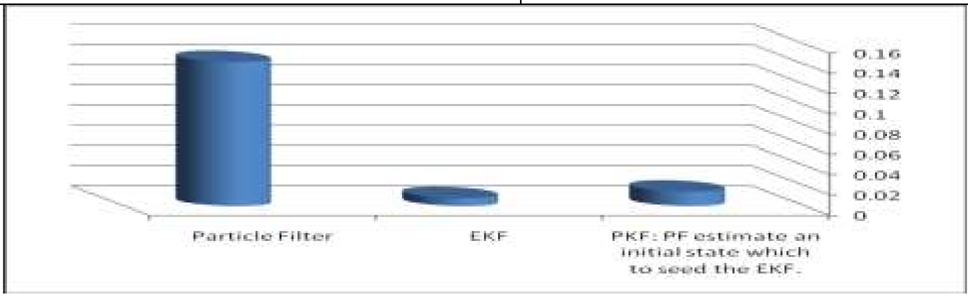


Fig.7: Comparing the time required to update the robot pose estimate after a range measurement is taken.

CONCLUSION

This paper presents a study for the effect of several filters in the behavior of robot localizer using radio beacons that provide the ability to measure range only. Different filters namely Kalman filter (KF), Particle Filter (PF) and a proposed enhanced particle/Kalman (EPKF) implemented in Matlab environment and their behavior are evaluated. The enhanced particle/ Kalman (EPKF) provide the required initial location while there is no significant change in the error in compared to Kalman filter (EKF) and computational cost.

Several approaches are reported to overcome the divergence of KF in case of (strong maneuvering and bad initial start. In this paper we purposed using PF as Initialization phase to coarsely predict the initial location.

REFERENCES

[1] D. Fox, W. Burgard, and S. Thrun, *Markov localization for mobile robots in dynamic environments,* Journal of Artificial Intelligence Research, 11:391–427,1999.
[2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, *Monte carlo localization: Efficient position estimation for mobile robots,* In Proceedings of the National Conference on Artificial Intelligence ,Orlando, FL, 1999.
[3] Ioannis M.Rekleitis, *A particle filter tutorial for mobile robot localization*, Technical Report TR-CIM-04-02,McGill University, Canada H3A 2A7,2004.
[4] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001
[5] James Carpenter and Peter Clifford and Paul Fearnhead "An Improved Particle Filter for Non-linear Problems" Department of Statistics, University of Oxford, 1999.
[6] Simon Maskell, Neil Gordon, and Tim Clapp, *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,* M. Sanjeev Arulampalam, IEEE Transactions on Signal Processing, Vol. 50, NO. 2,FEBRUARY 2002.
[7] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, *Robust Monte Carlo localization for mobile robots, Artificial* Intelligence, pp. 99-141, 2000.
[8] J. Liu and R. Chen., *Sequential Monte Carlo methods for dynamic systems,* Journal of American Statistical Association, 93, 1998.
[9] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
[10] J. Z. Sasiadek and P. Hartana, *Sensor data fusion using Kalman filter*, InProceedings of the Third International Conference on Information Fusion, vol. 2,pp. 19-25, 2000.].
[11] Joseph Djugash, and Sanjiv Singh "*A Robust Method of Localization and Mapping Using Only Range* "The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA 2009
[12] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, 2001.

[13] Howida A. El-Halym, I. I Mahmoud, A. AbdelTawab, and S. E.-D. Habib, *Appraisal of an Enhanced Particle Filter for Object Tracking,* Proceedings of IEEE International conference on Image processing (ICIP) Conf., Cairo, Egypt, Nov. 2009, pp. 4105-4107
[14] B. Ristics, S. Arulanpulam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, 2004.
[15] M. Isard and A. Blake. Condensation *–, conditional density propagation for visual tracking,* International Journal of Computer Vision 29(1), pp. 5—281998.
[16] I. I Mahmoud, A.A. Tawab, M. Salama, H.A.A. El-Halym, *Appraisal of Different Particle Filter Resampling Scheme Effect in Robot Localization*,29th NATIONAL RADIO SCIENCE CONFERENCE (NRSC 2012)
[17] Derek Kurth , *Range-Only Robot Localization and SLAM with Radio,* Robotics Institute, Carnegie Mellon Univ., tech. report CMU-RI-TR-04-29 May 2004.